

# Operating the DG8SAQ VNWA from within Linux using Wine

T. Baier DG8SAQ

December 28<sup>th</sup> 2017

## Introduction

As to date the VNWA software is only available for Microsoft Windows platforms, a Windows-like platform needs to be built inside Linux in order to use it. Up to now, this could only be achieved by installing a real Windows operating system inside a virtual machine running on the Linux platform or by running VNWA inside Wine. As Wine still does not support a USB stack, the previous VNWA Wine package circumvented this problem by rerouting the calls VNWA dispatches to the USB driver to a virtual serial com port created with socat, which hooked up to the Linux side. Unfortunately, with the advent of Wine2.x, there are problems connecting Wine to socat. Therefore, the present package implements communication via TCP instead of a virtual serial port. This makes socat as intermediate communication layer unnecessary and thus improves the overall simplicity and stability of the solution.

No special solution is required to connect to the USB audio codec(s) since Wine has already built-in audio support.

The software has been tested on Debian 9 Stretch 32 bit and Ubuntu14.04 64bit.

Mandatory software packages:

- Wine (tested with version 2.0, older versions are fine, too.)
- VNWA installer version 36.7.4 or newer (includes the described Linux package and documentation)

Some systems may require installation of additional packages for `usb_lan_server` to execute, e.g. `libusb-dev`.

## Installation

The installation process is described for a Debian Linux system. On other systems use an appropriate package manager, e.g. YAST on OpenSUSE.

This is a step by step installation procedure which tests functionality in intermediate steps.

1. Install wine:            In a root terminal do:            `apt-get install wine`
2. Download the VNWA software installer for Windows (VNWA-installer.exe) and install the VNWA software inside wine skipping driver and certificate installation:  
`wine VNWA-Installer.exe`

Important: You need at least VNWA version 36.7.4. The latest installer version can be found here:

[https://groups.yahoo.com/neo/groups/VNWA/links/all/A1\\_DG8SAQ\\_001284012495](https://groups.yahoo.com/neo/groups/VNWA/links/all/A1_DG8SAQ_001284012495)

In the following it is assumed that VNWA was installed into `c:\VNWA`, which is found in the Linux file system as `$HOME/.wine/drive_c/VNWA`.

If you use a different directory, you must later manually change the paths in `VNWA.sh`.

- The installer also copies the Linux package to your disk. The Linux files are found in `$HOME/.wine/drive_c/VNWA/Linux` .  
Copy the following Linux specific installation file provided in the Linux folder to the VNWA installation folder:

- `LibUSB0.dll`

Copy the most appropriate versions of files found in subfolders of the Linux folder to the Linux folder `$HOME/.wine/drive_c/VNWA/Linux` :

- `usb_lan_server`

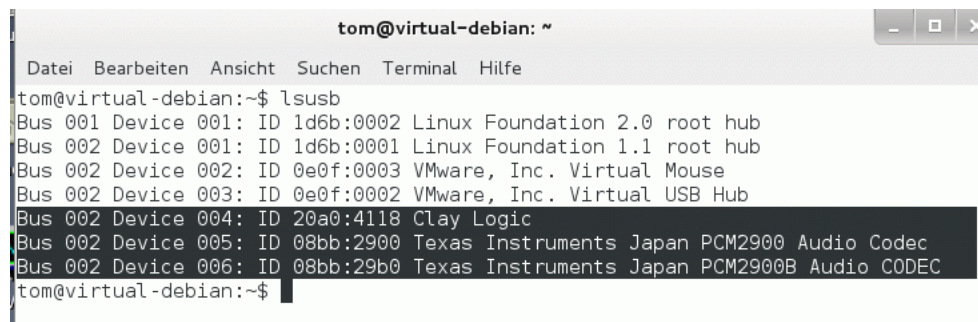
- `usb_lan_testclient`

( make sure the programs are executable: `chmod +x usb_lan_*` )

Make sure the script `VNWA.sh` in `$HOME/.wine/drive_c/VNWA/Linux` is executable(`chmod +x VNWA.sh` ).

The file `$HOME/.wine/drive_c/VNWA/Linux/AEA9_VNWA.0.png` is the icon for later use on the desktop.

- Connect the VNWA and check if Linux can see all devices. In a console type `lsusb`



```
tom@virtual-debian: ~
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
tom@virtual-debian:~$ lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 002 Device 002: ID 0e0f:0003 VMware, Inc. Virtual Mouse
Bus 002 Device 003: ID 0e0f:0002 VMware, Inc. Virtual USB Hub
Bus 002 Device 004: ID 20a0:4118 Clay Logic
Bus 002 Device 005: ID 08bb:2900 Texas Instruments Japan PCM2900 Audio Codec
Bus 002 Device 006: ID 08bb:29b0 Texas Instruments Japan PCM2900B Audio CODEC
tom@virtual-debian:~$
```

You should see the VNWA controller with ID 20a0:4118 and one (VNWA2, VNWA3) or two (VNWA3E) PCM2900 USB audio devices as shown above.

- Normally, only root has raw access to USB devices. So, next, we enable raw USB access for a standard user.

In a root terminal or “sudo” do...

- 5.1. Create a group named usb: `addgroup --system usb`

If the group already exists, you will see an error message. This is ok.

- 5.2. Put yourself into this group: `adduser [your username] usb`

If you are already a group member, you will see an error message. This is ok.

- 5.3. Create the file `/etc/udev/rules.d/80-usb.rules`

and add the following single line to it using an editor like nano:

```
SUBSYSTEM=="usb" ATTRS{idVendor}=="20a0"
```

```
ATTRS{idProduct}=="4118" GROUP="usb" MODE:="0660"
```

You may instead simply copy the file `80-usb.rules` provided in the Linux package.

- 5.4. Reboot your Linux system or alternatively run:

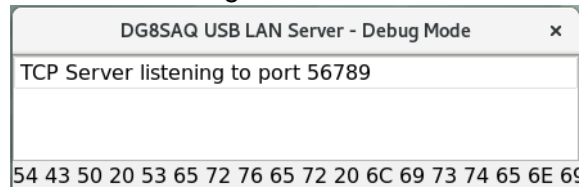
```
udevadm control --reload-rules && udevadm trigger
```

6. Now, we test if we can manually access the VNWA controller using the programs `usb_lan_server` and `usb_lan_testclient` previously copied to the `$HOME/.wine/drive_c/VNWA/Linux` directory. Make sure both are executable.

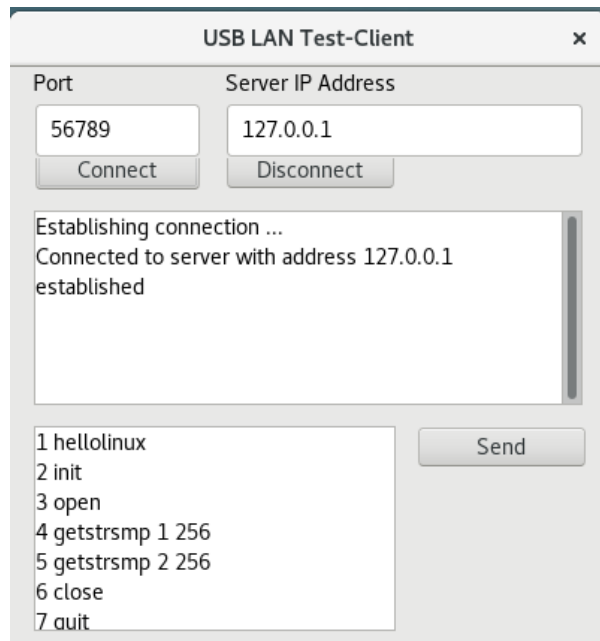
```
tom@virtualdebian9: ~/.wine/drive_c/VNWA/Linux
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
tom@virtualdebian9:~/.wine/drive_c/VNWA/Linux$ chmod +x usb_lan_*
tom@virtualdebian9:~/.wine/drive_c/VNWA/Linux$ ls -l usb_lan*
-rwxr-xr-x 1 tom tom 2078688 Nov  9 11:31 usb_lan_server
-rwxr-xr-x 1 tom tom 2848712 Nov  9 09:27 usb_lan_testclient
tom@virtualdebian9:~/.wine/drive_c/VNWA/Linux$ ./usb_lan_server -d &
[1] 7724
tom@virtualdebian9:~/.wine/drive_c/VNWA/Linux$ ./usb_lan_testclient
```

Now, run `usb_lan_server -d &` (-d to get the debug screen, & to detach the process from the console) and `usb_lan_testclient` as shown above.

The server debug screen should indicate that the server is open and listening:

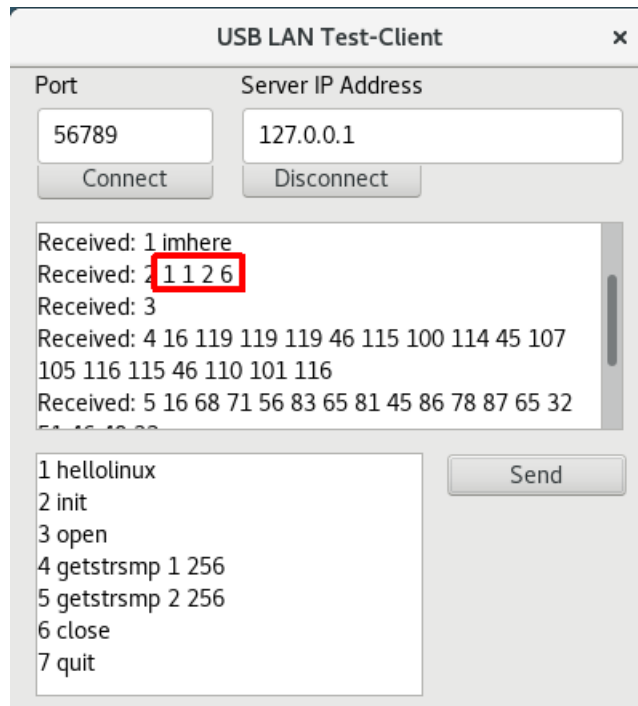


The test client windows should look like this after pressing the `Connect`-button:



Also, the server should notify that the client is connected now.

After the connection is established, press the client's send button to send the preloaded commands to the server. The server should in turn mirror every command and answer and the client should receive the server's answers:



Particularly interesting is the answer to the `init` command (1 1 2 6 above, marked red) which indicates the version of the libusb-driver (here 1.1.2.6). This may vary from system to system, but if the answer is 0 0 0 0, then there is no VNWA connected or you do not have raw USB access. You may see additional error messages in this case. Also, you may check if missing raw access is causing this problem by running the same programs as root. If successful as root, then you are not granted raw access as a simple non-privileged user.

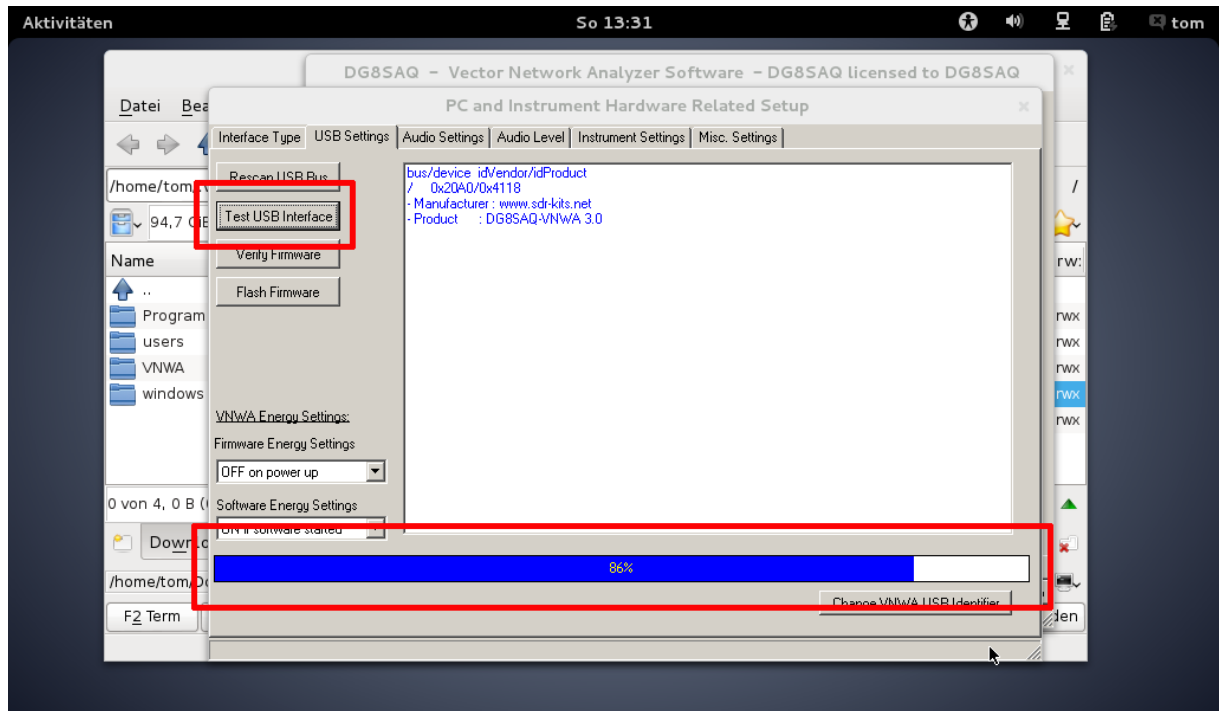
7. Launch the VNWA software by executing  
`$HOME/.wine/drive_c/VNWA/Linux/VNWA.sh`  
 Note, that on my web book it takes about 10 seconds for VNWA to come up.
8. In VNWA go to Options-Setup USB Settings, make sure your VNWA license code is entered correctly and press "Rescan USB Bus". The result should look like this:



This proves that the VNWA software can connect to the VNWA hardware.

Note, that in this Linux implementation, the “Rescan USB Bus” feature will always show at most one VNWA device with reduced information (see above). So, only one VNWA may be connected to the computer at the same time.

9. You may check the USB communication including speed by using the “Test-USB Interface” feature in the USB setup:

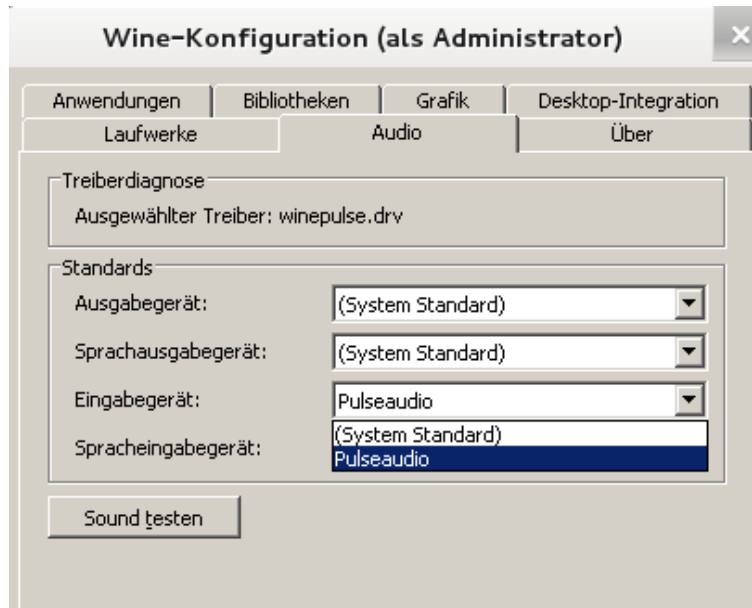


On my Windows7 HP ProBook the test runs through in about 1 second, while the same test takes about 10 seconds on my Debian Wheezy Acer AspireOne. This is still sufficient to operate the VNWA.

We are not yet ready to measure, but as a further test, you may check if you can control the output frequency of your VNWA in signal generator mode.

10. Next, we have to take care of the audio codecs in Wine, therefore terminate the VNWA application.

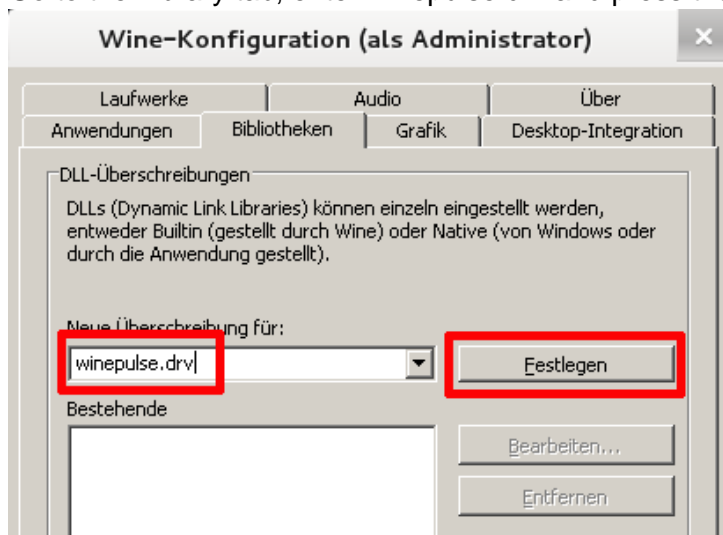
11. In a Linux console run `winecfg` and select the Audio tab:



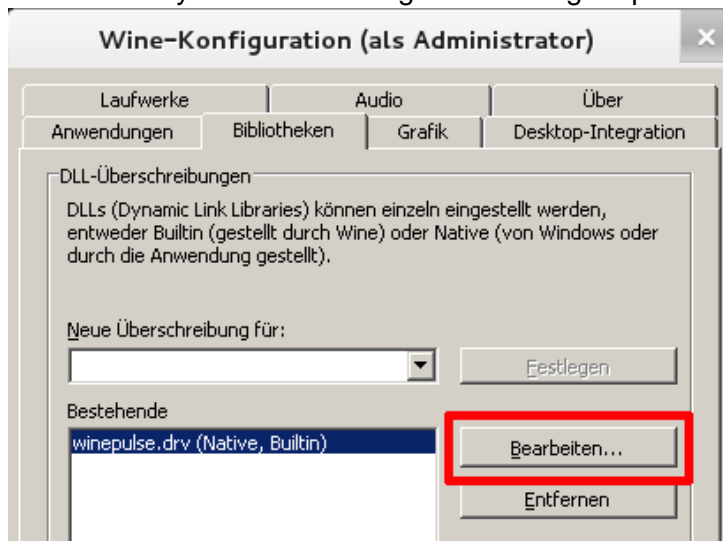
On my Debian machine, Wine 1.6.2 and Wine 1.7.2 and Wine 2.0 refused to detect any of the VNWA USB sound devices, but only showed a device named Pulseaudio:

This can be resolved by deactivating PulseAudio in Wine:

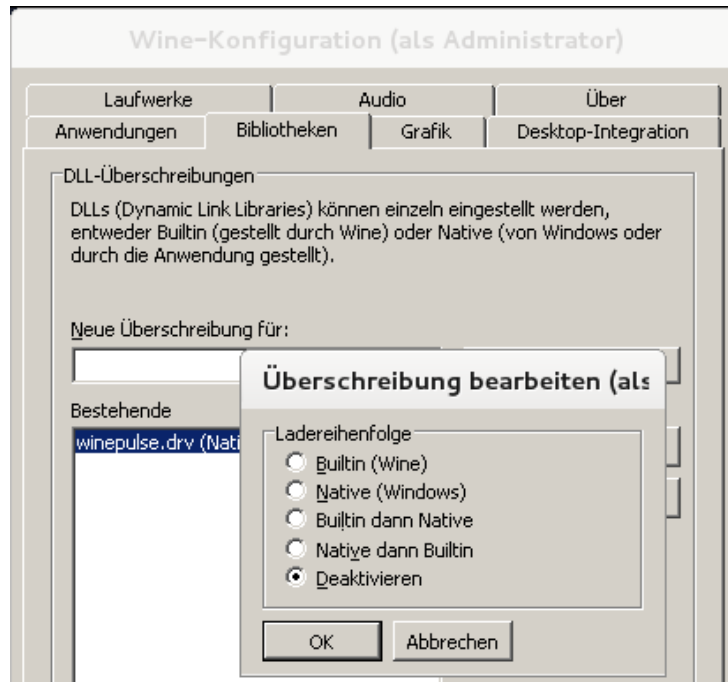
12. Go to the Library tab, enter `winepulse.driv` and press the apply button:



Confirm that you want to change the loading sequence. Next, press the Edit button,...

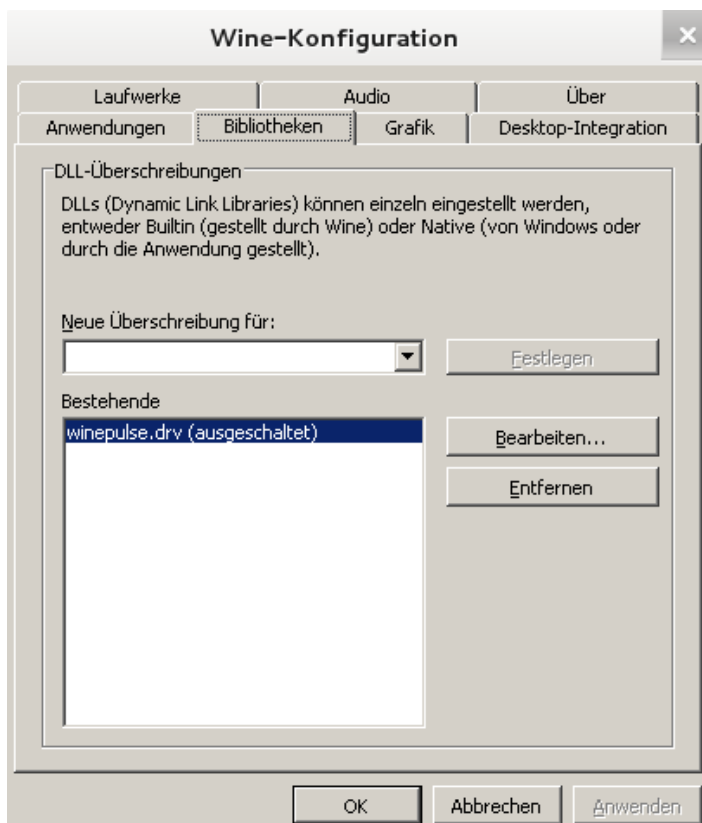


... select "deactivate" ...

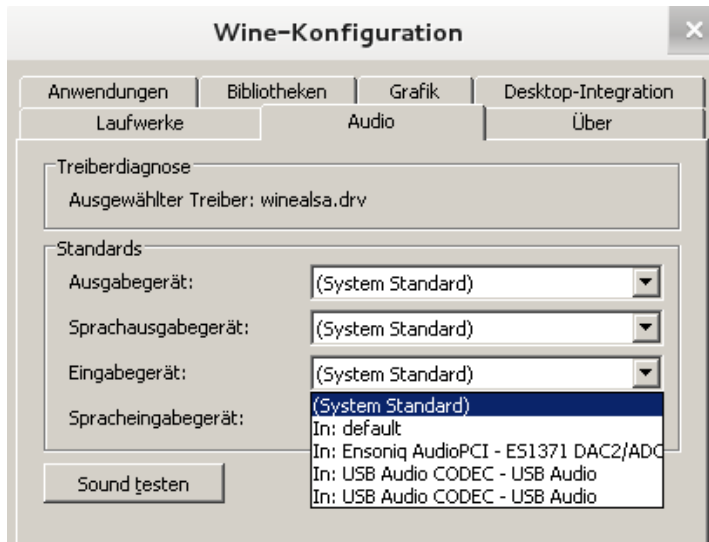


And confirm by pressing the OK button.

Now, winepulse should show as deactivated:



Close winecfg by pressing the OK button, restart winecfg and observe that all input devices are visible now:



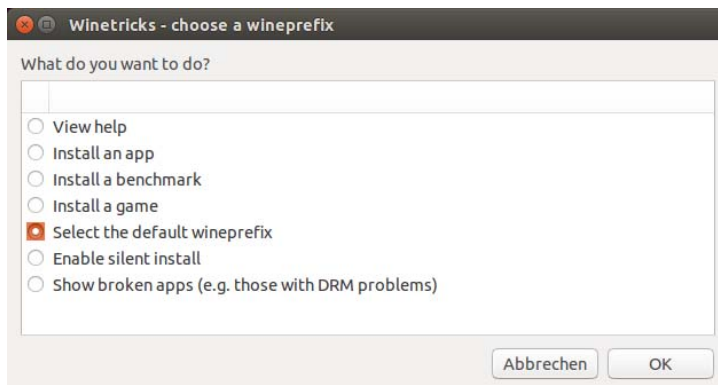
Close winecfg again.

On an Ubuntu 14.04 64bit system the audio codecs were still not visible in Wine. Here, the following has worked in order to launch ALSA support in Wine:

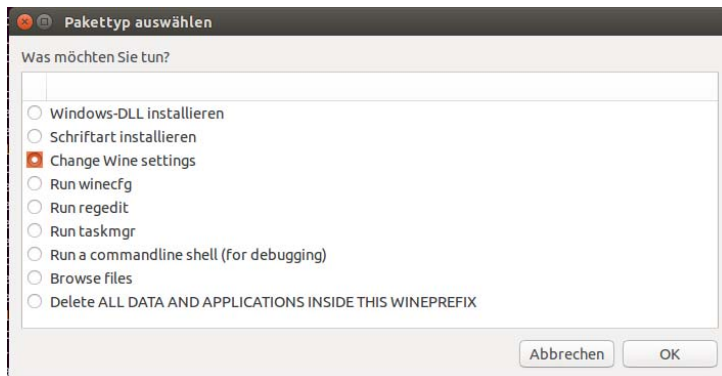
In a terminal launch

```
winetricks
```

and check "Select the default wineprefix":

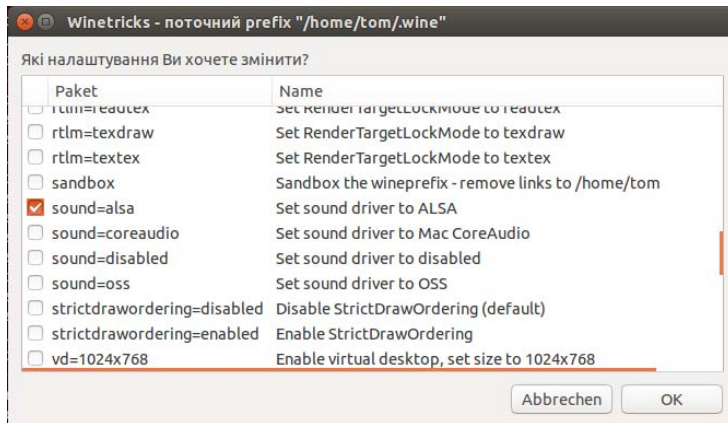


Press "OK" and then check "Change Wine settings":



Press "OK" and scroll down to and select "sound=alsa":





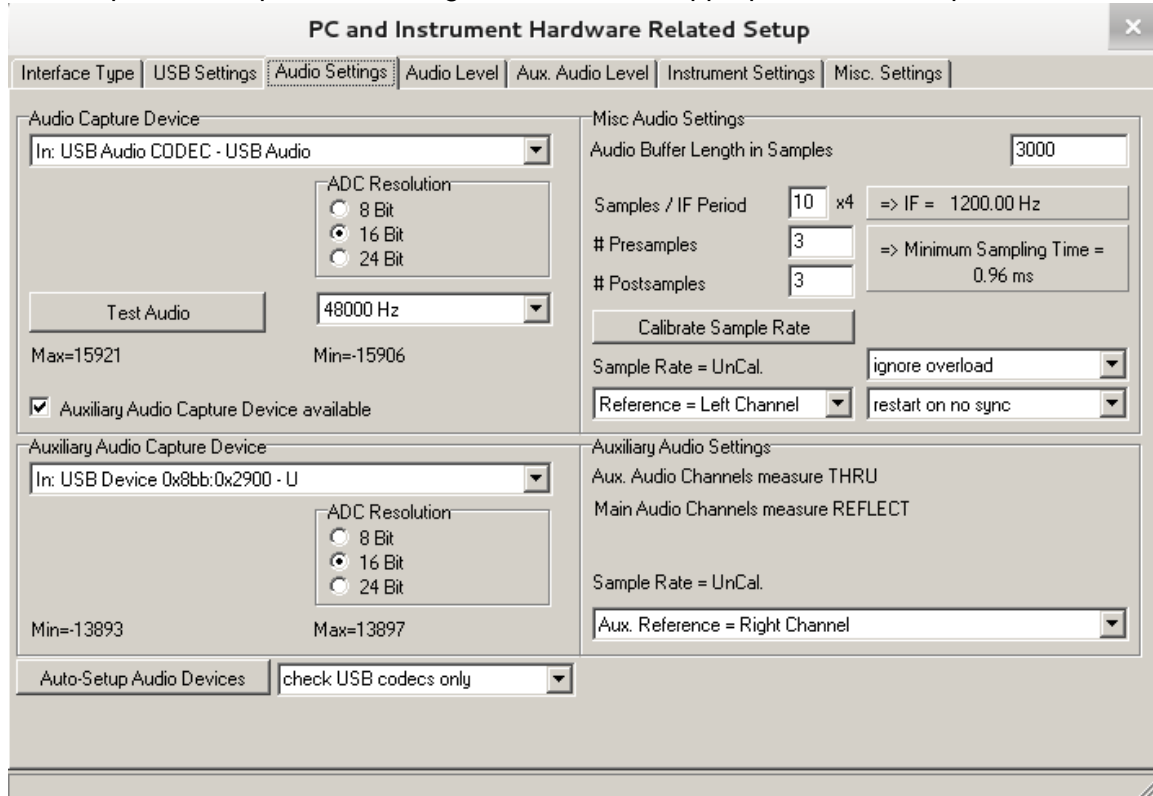
Press "OK" and then close Winetricks by clicking the "x".

Launch `winecfg` and you should see the VNWA USB audio codecs now.

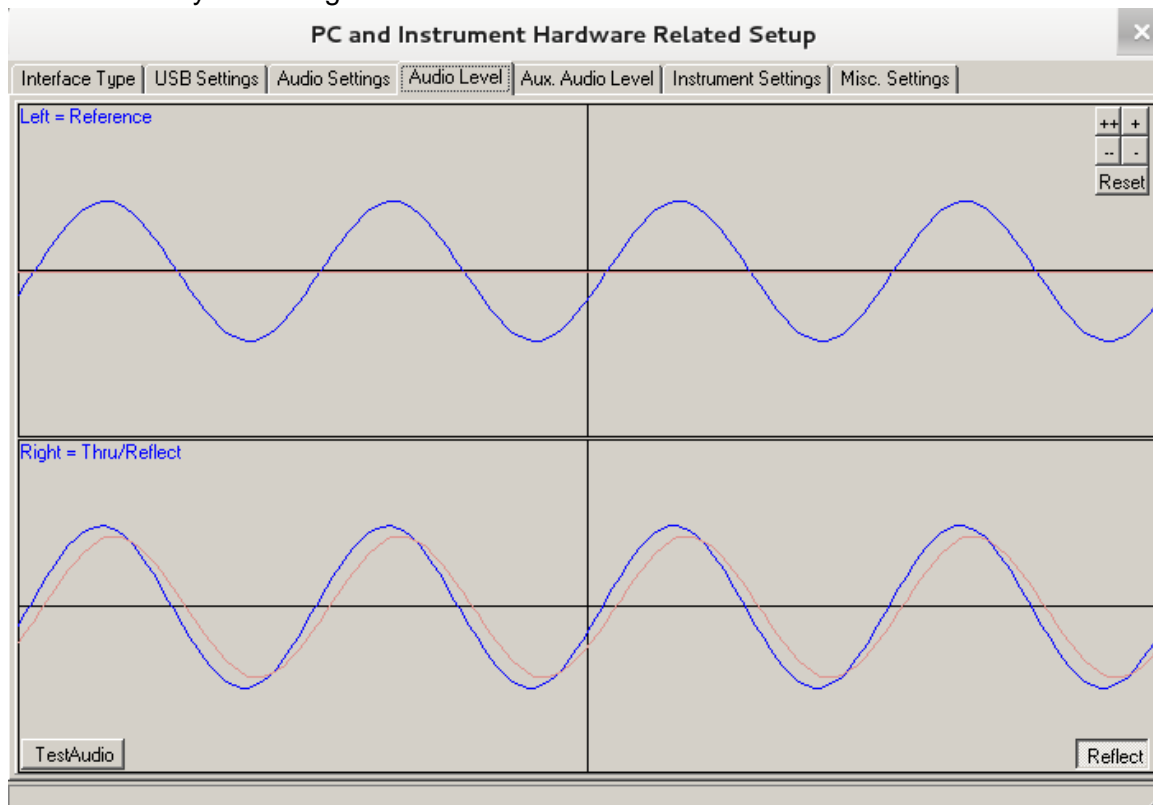
Most interestingly, using above trick makes deactivating pulseaudio in Wine unnecessary.

We are ready now to set up VNWA.

13. Launch VNWA again by running `./VNWA.sh` inside the Linux folder of the VNWA installation directory.  
Go to Options-Setup-Audio Settings and select the appropriate Audio Capture Devices:



Press "Test Audio" and check if you can open the selected Audio devices. Check in Audio Level if you see signals from the VNWA:



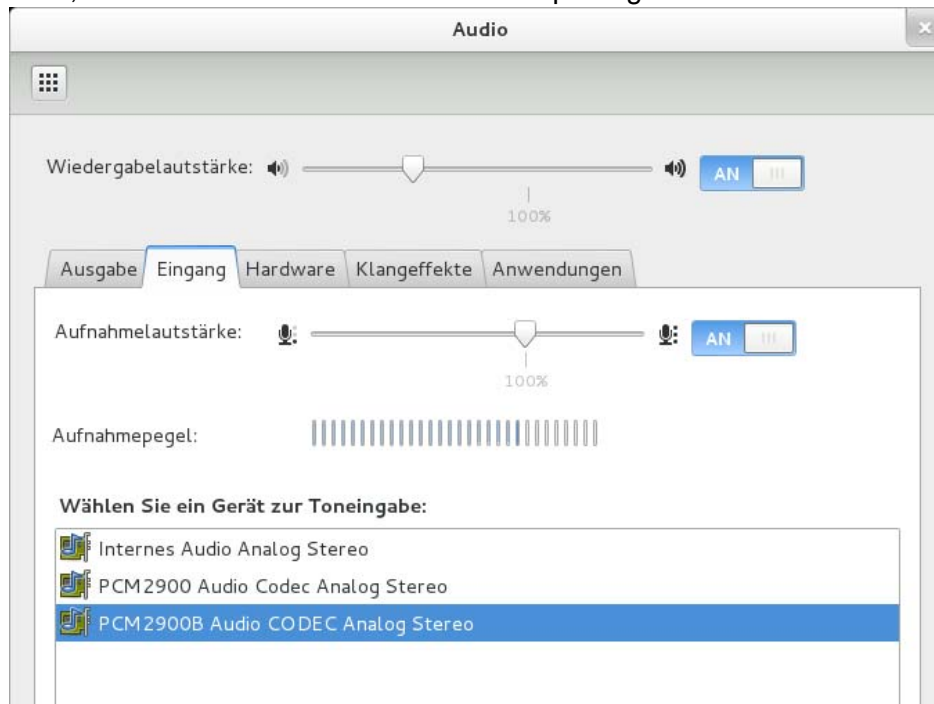
Note, that above sine waves may also be triangular depending on audio settings. This is ok.

14. As the VNWA3E normally has two identical USB sound chips built-in, accessing these from Wine poses a bit of a challenge. Since they share the same name, VNWA inside Wine can only access one of them (unless by chance you have two different PCM2900 subtypes built-in). There is still a possibility to access both of them simultaneously.

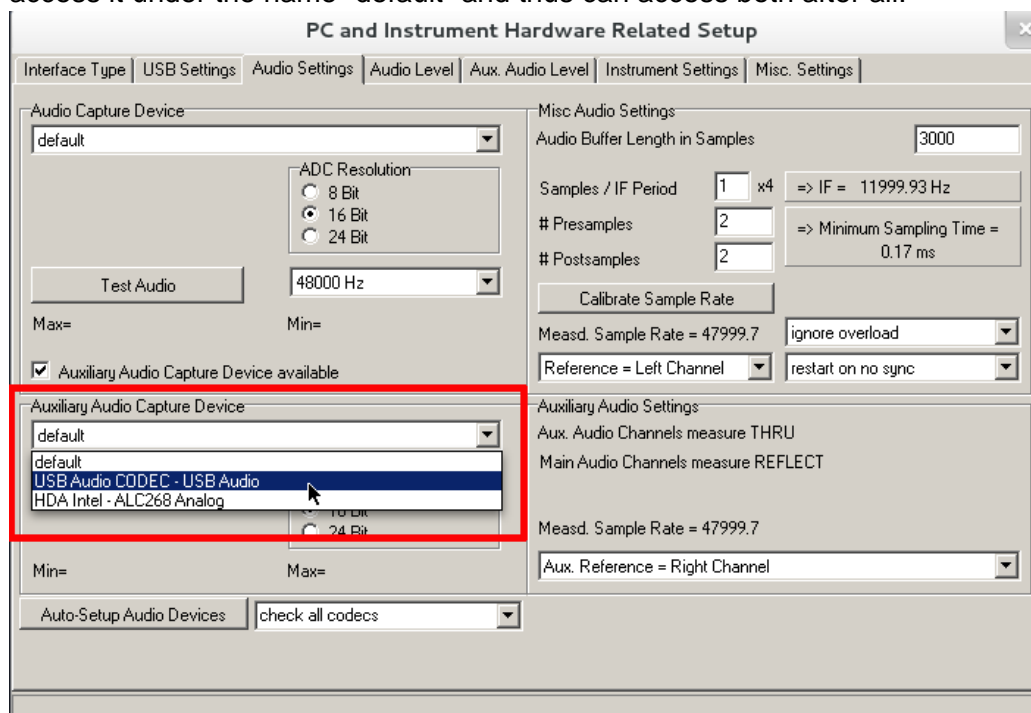
a) Quick and dirty:

First find out, which codec VNWA can access directly in Wine using the AudioLevel feature in VNWA setup.

Next, make the other codec the default capturing device in Linux:



By setting the unseen USB codec to be the default one as seen above, Wine can access it under the name "default" and thus can access both after all:



b) The sophisticated way (many thanks to Michael for providing this method!)

It is possible to alias sound devices inside Wine with unique names:

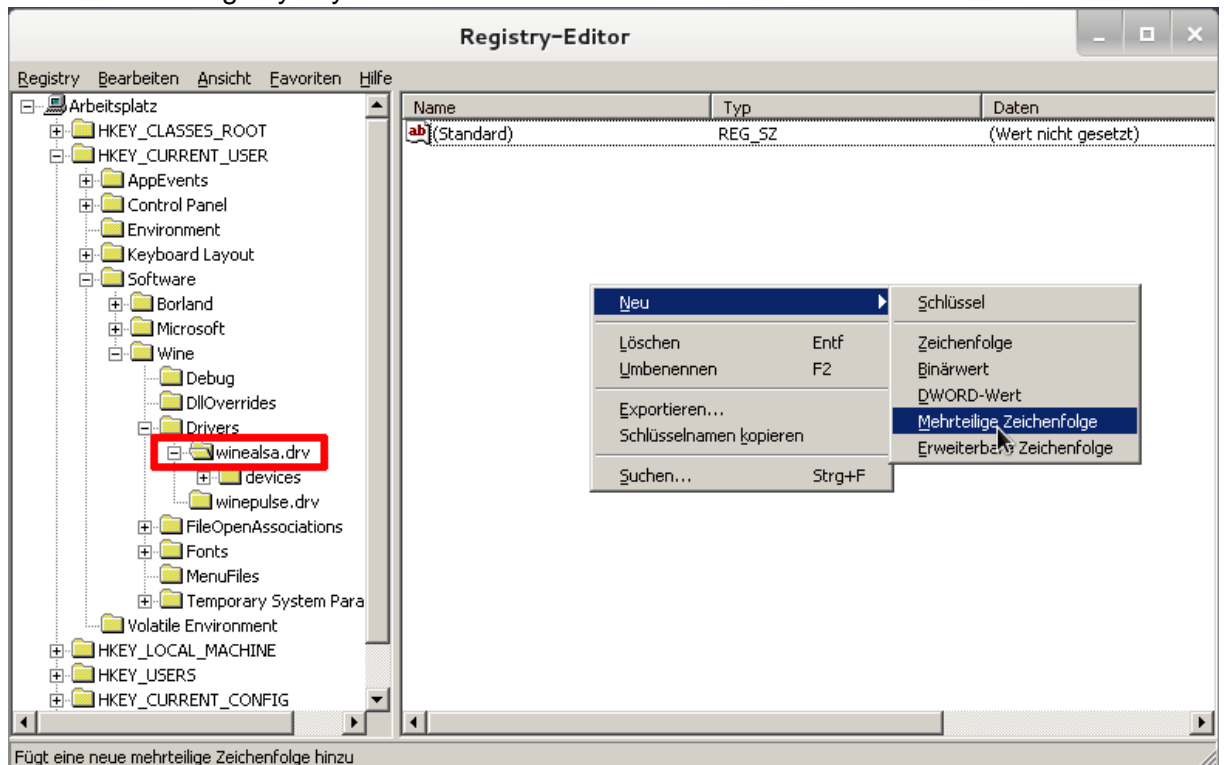
- i. In a Linux terminal window list your Alsa device config by executing the command `arecord -l`

```
tom@virtual-debian7: ~  
Datei Bearbeiten Ansicht Suchen Terminal Hilfe  
tom@virtual-debian7:~$ arecord -l  
**** Liste der Hardware-Geräte (CAPTURE) ****  
Karte 0: AudioPCI [Ensoniq AudioPCI], Gerät 0: ES1371/1 [ES1371 DAC2/ADC]  
  Sub-Geräte: 1/1  
    Sub-Gerät #0: subdevice #0  
Karte 1: CODEC [USB Audio CODEC], Gerät 0: USB Audio [USB Audio]  
  Sub-Geräte: 1/1  
    Sub-Gerät #0: subdevice #0  
Karte 2: CODEC_1 [USB Audio CODEC], Gerät 0: USB Audio [USB Audio]  
  Sub-Geräte: 1/1  
    Sub-Gerät #0: subdevice #0  
tom@virtual-debian7:~$
```

More details can be obtained by running `cat /proc/asound/cards --`

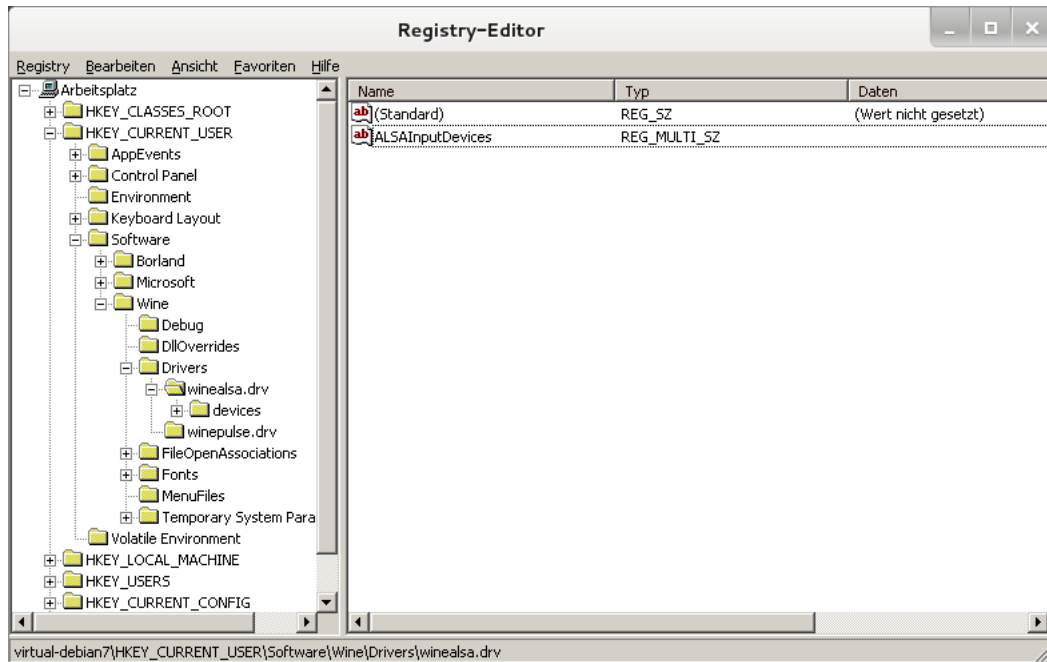
- ii. In my case the two VNWA USB codecs are found as `card1/subdevice0` (name CODEC) and `card2/subdevice0` (name CODEC\_1) respectively. We do need these numbers or names for identification in order to create valid ALSA addresses for the two VNWA codecs.
- iii. Next, inside the Linux terminal execute `wine regedit`

Browse to the registry key shown below...

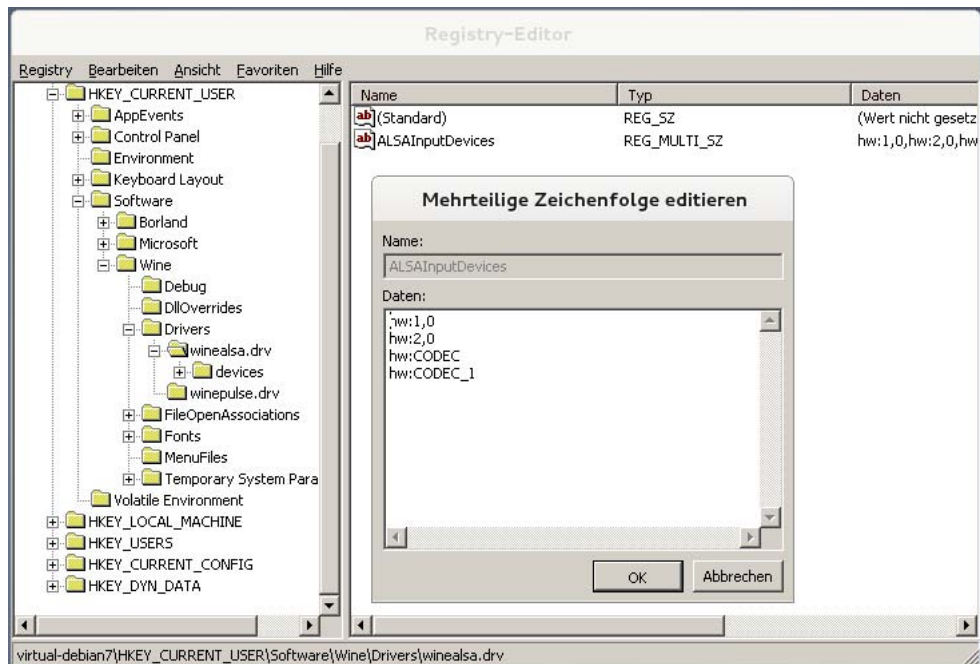


... and add a multipart character sequence entry with name **ALSAInputDevices** by right-clicking into the right white field of key **winealsa.driv**.

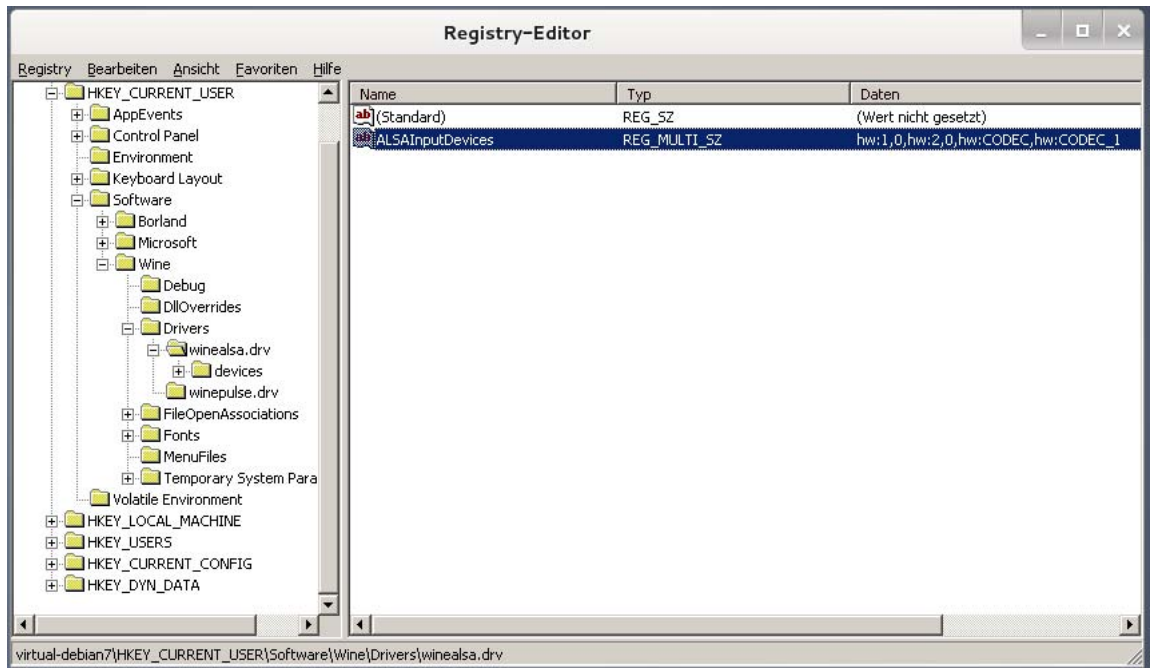
The result should look like this:



Double-click the entry `ALSAInputDevices` and enter valid ALSA addresses (hw:card number, subdevice number) of the VNWA codecs containing the numbers previously found with `arecord -l`:

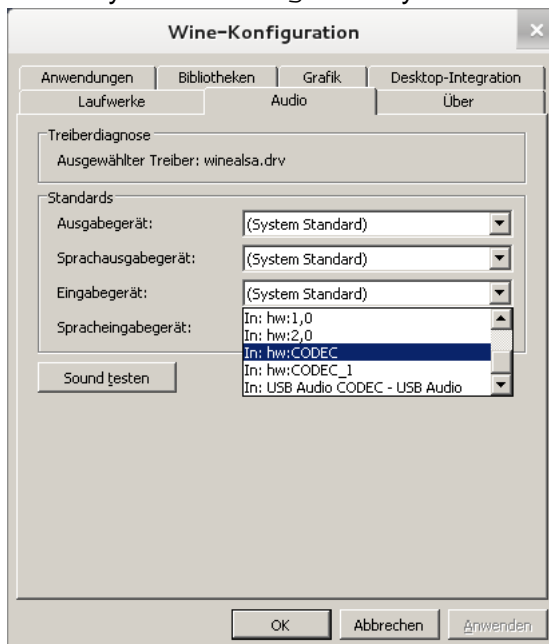


Press OK when done. The result should look like this:



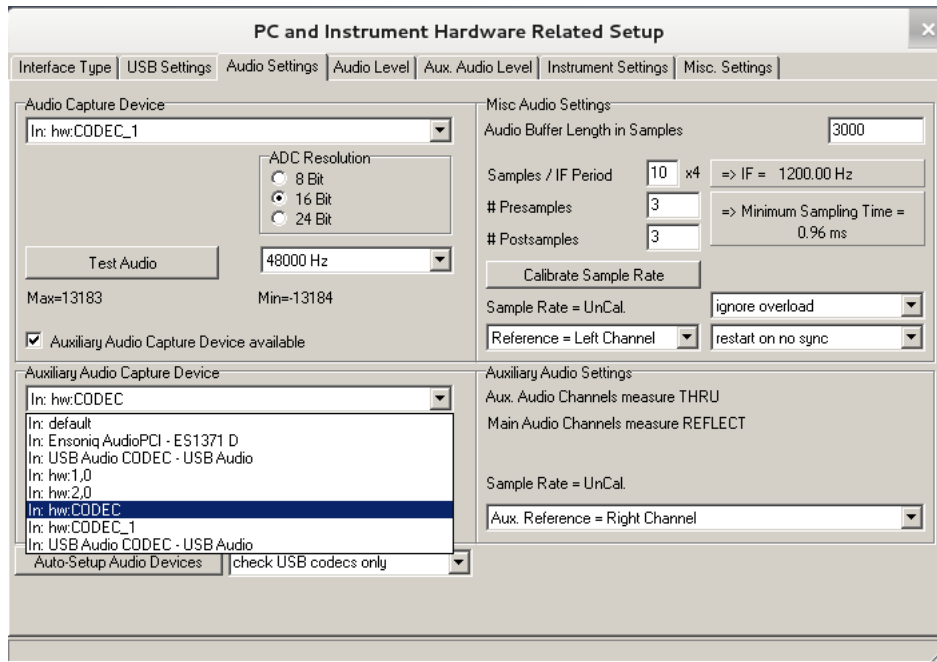
Close the registry editor.

- iv. You may run `winecfg` to verify that the renaming has worked:



Now, indeed four additional input devices with names `hw:1,0`, `hw:2,0`, `hw:CODEC` and `hw:CODEC_1` are available. The two VNWA3E USB codecs may be accessed via these unique names. If you later add or remove audio codecs to your system, the card numbers may change. Thus the `hw:CODEC*` devices may be superior as they may also be found after the system audio configuration has changed.

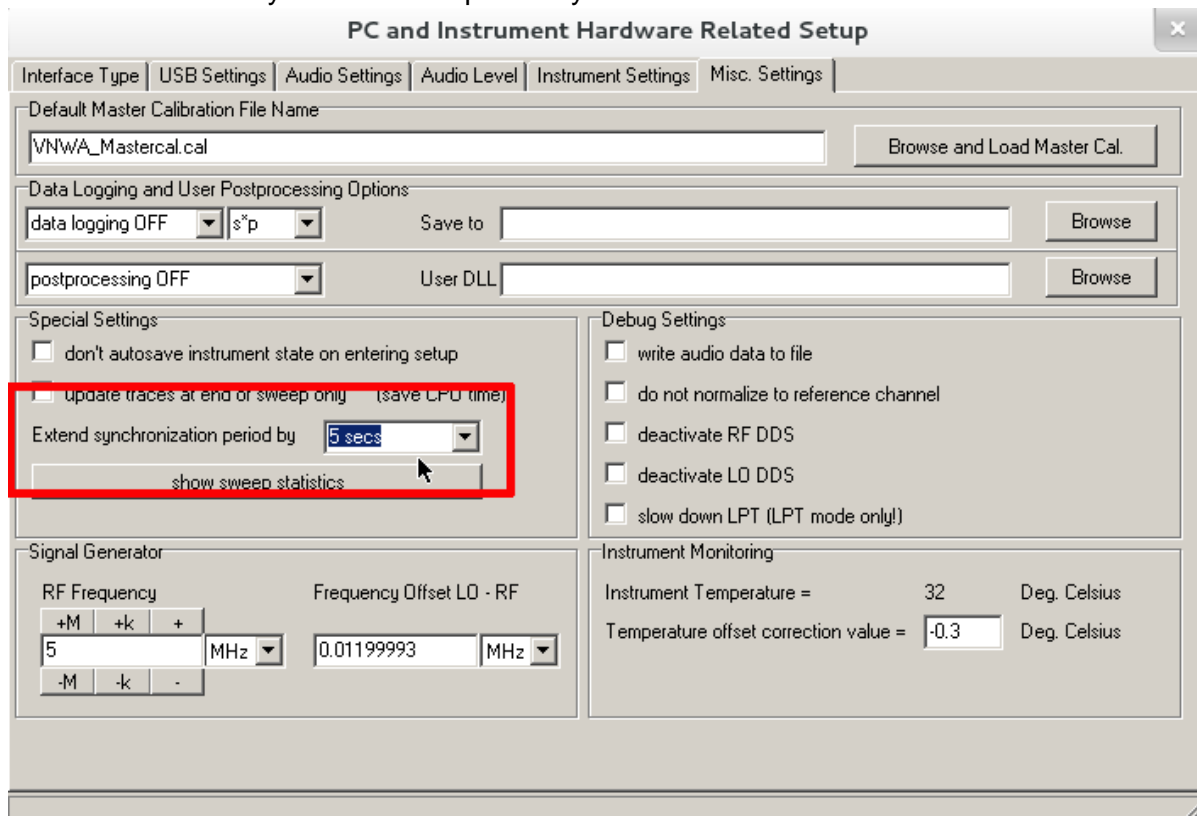
- v. Launch VNWA, go to Setup-Audio Settings and find and select the new sound devices there:



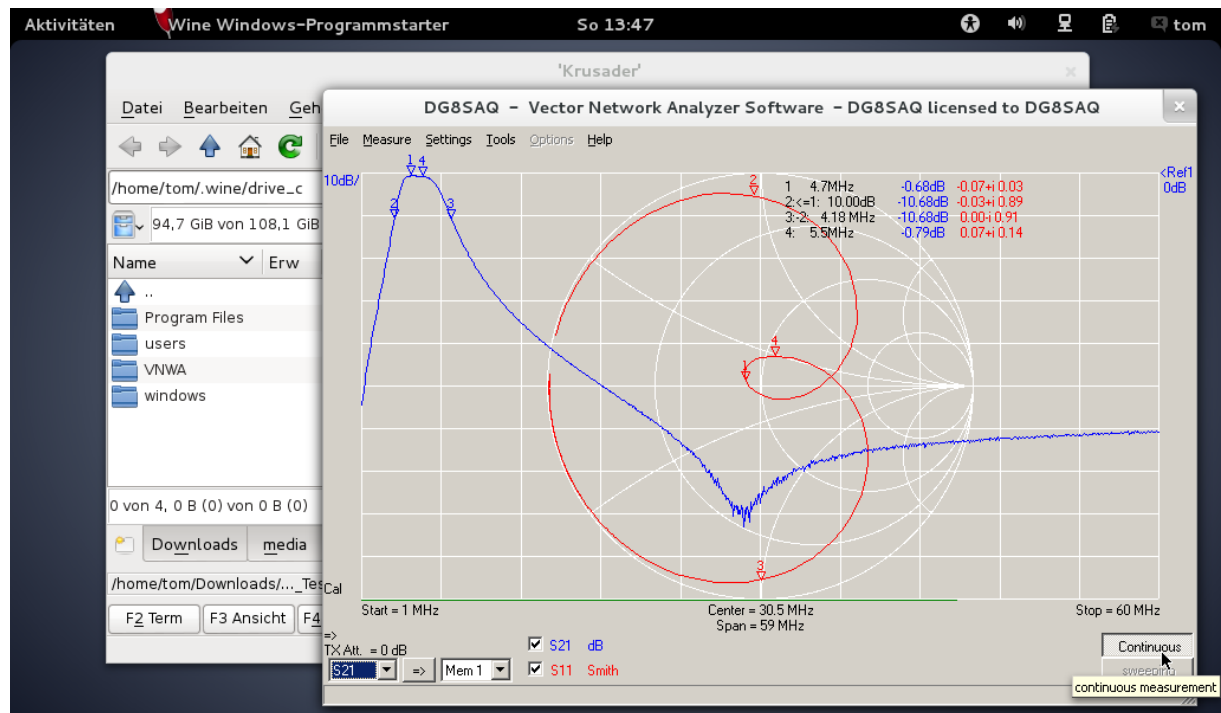
From my experience, it appears that hw:2,0 (or CODEC\_1, i.e. the device with the highest card number) is always the main codec independent of the USB port used for the VNWA, but better double-check using the AudioLevel tab. The main codec is the one where one signal changes when toggling between Thru and Reflect.

15. Now, do a normal setup as described in the VNWA documentation.

16. Extend the VNWA synchronization period by 5 seconds as shown below:



Now, the VNWA system is ready to do measurements:



## 17. Adding a desktop icon for VNWA:

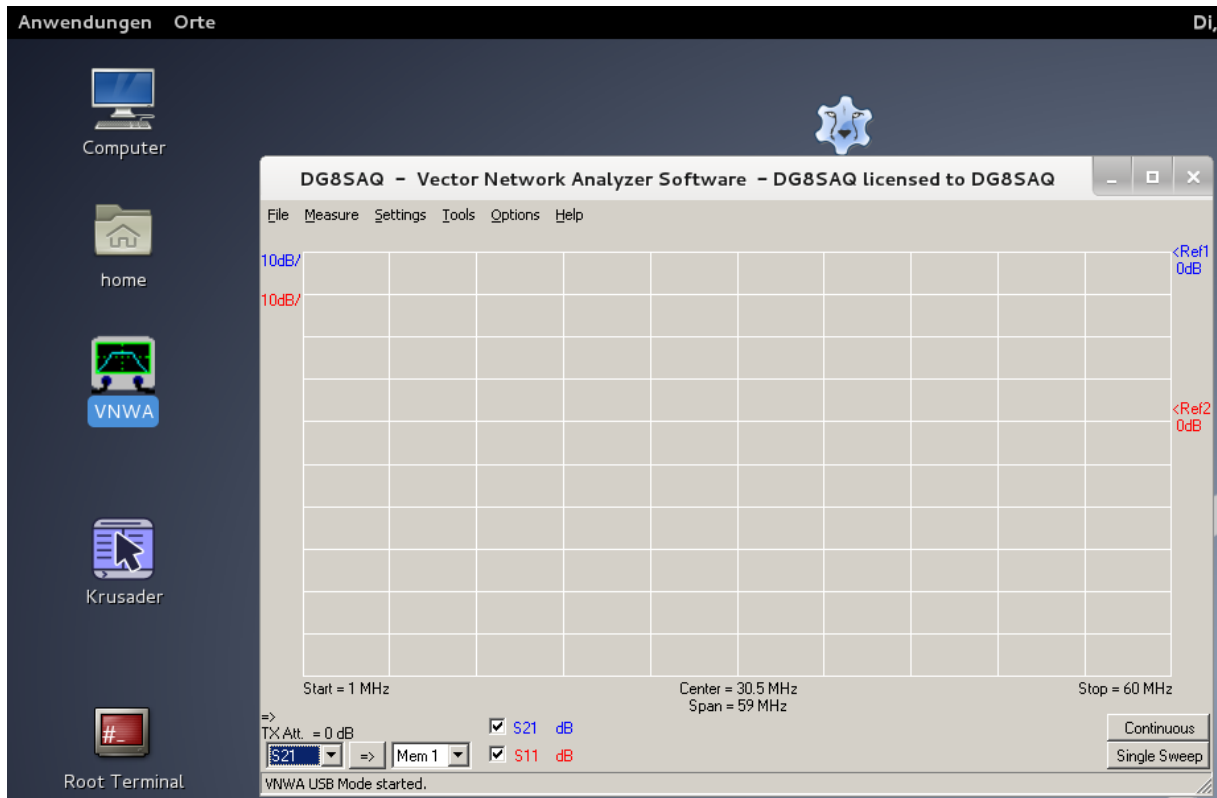
Create the file `VNWA.desktop` in your desktop directory, typically `$HOME/desktop` (or `$HOME/Arbeitsfläche` for German systems) with the following contents (replace `tom` by your own username, you may use the file template from the Linux package):

```
[Desktop Entry]
Version=1.0
Type=Application
Name=VNWA
Comment=VNWA
Exec=/home/tom/.wine/drive_c/VNWA/Linux/VNWA.sh
Path=/home/tom/.wine/drive_c/VNWA/Linux
Icon=/home/tom/.wine/drive_c/VNWA/Linux/AEA9_VNWA.0.png
Terminal=false
StartupNotify=true
GenericName=VNWA
```

Make the file executable: `chmod +x VNWA.desktop`

Now you should see an icon on your desktop with which you can launch the VNWA application:





You may have to make desktop icons visible on your Linux system using the `dconf-editor`.

END